



Your complimentary use period has ended.
Thank you for using PDF Complete.

Click Here to upgrade to Unlimited Pages and Expanded Features

Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan guna Memperoleh Gelar Sarjana Sains



Oleh
Yunidha Ekasari
NIM 11305141002

PROGRAM STUDI MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA
2015

i

BAB I

PENDAHULUAN

A. Latar Belakang

Kanker merupakan salah satu penyakit yang paling berbahaya dan mematikan. Kanker dapat menyerang siapapun tanpa mengenal usia dan jenis kelamin seseorang. Kanker yang terdapat pada serviks atau leher rahim, yaitu area bawah rahim yang menghubungkan rahim dengan vagina disebut kanker serviks. Kanker serviks merupakan jenis kanker terbanyak yang diderita kaum wanita setelah kanker payudara.

Menurut *National Cancer Institute*, faktor-faktor penyebab kanker serviks adalah infeksi *Human Papillomavirus* (HPV) yakni HPV 16, HPV 18, HPV 31 HPV 33 dan HPV 45, perilaku seksual (seks usia dini dan bergonta-ganti pasangan), rokok (wanita yang merokok sekitar dua kali lebih mungkin terjangkit kanker serviks) serta perempuan yang mengonsumsi obat Dietilstilbestrol (DES) selama kehamilan untuk mencegah keguguran.

Kanker serviks sering disebut sebagai “silent killer” sebab pada awal kemunculannya sulit sekali terdeteksi atau tanpa gejala tetapi gejala-gejala kanker akan muncul jika kanker sudah memasuki tahap lanjut. Menurut dr.Andrijono selaku ketua penyelenggara pertemuan *Asia Oceania Research Organization in Genital Infection & Neoplasia* (AOGIN) di Kuta Bali, “Di Indonesia, lebih dari 70 persen kasus kanker serviks ditemukan saat sudah stadium lanjut (di atas stadium 2)”. Hal ini dapat dicegah apabila kanker serviks terdeteksi lebih dini. Berbagai

DIAGNOSIS KANKER SERVIKS MENGGUNAKAN MODEL RECURRENT NEURAL NETWORK (RNN) BERBASIS GRAPHICAL USER INTERFACE (GUI)

Oleh:
Yunidha Ekasari
NIM 11305141002

ABSTRAK

Lebih dari 70% kasus kanker serviks terdeteksi ketika sudah memasuki stadium lanjut. Salah satu cara deteksi dini kanker serviks adalah pemeriksaan kolposkopi. Melalui citra kolposkopi servik dapat diperoleh berbagai informasi guna mendiagnosis kanker serviks. Pada penelitian ini menggunakan model *Recurrent Neural Network* (RNN) untuk mendiagnosis kanker serviks berdasarkan citra kolposkopi serviks. Penelitian ini bertujuan untuk menjelaskan aplikasi dan proses membangun *Graphical User Interface* (GUI) untuk diagnosis kanker serviks menggunakan model RNN berdasarkan citra kolposkopi serviks serta mendeskripsikan tingkat sensitivitas, spesifisitas dan akurasi diagnosis kanker serviks menggunakan model RNN.

Proses yang dilakukan adalah mengubah citra *red green blue* (RGB) ke citra *grayscale* pada data citra kolposkopi serviks. Setelah itu dilakukan pengekstrakan citra menggunakan metode *Gray Lavel Co-occurrence Matrix* (GLCM) diperoleh 14 fitur. Fitur tersebut digunakan sebagai variabel *input* sedangkan target merupakan kondisi serviks dari masing-masing citra. Data dibagi menjadi 2 bagian yakni data *training* dan data *testing*. Setelah itu dilakukan pelatihan RNN menggunakan jaringan Elman dan algoritma *backpropagation* dengan normalisasi data, perancangan model terbaik dan denormalisasi data. Perancangan model RNN terbaik dilakukan dengan mencari banyak neuron pada lapisan tersembunyi, membandingkan nilai MSE dan menggunakan prinsip parsimoni.

Model terbaik RNN yang diperoleh adalah 13 fitur *input* (*energy, contrast, correlation, sum of square, IDM, sum average, sum variance, sum entropy, entropy, difference entropy, maximum probability, homogeneity dan dissimilarity*), 7 neuron pada lapisan tersembunyi, dengan fungsi aktivasi *sigmoid bipolar* pada lapisan tesembunyi dan *linear* pada lapisan *output*. Kemudian dirancang sebuah program berbasis GUI dari model RNN terbaik. Program tersebut akan menampilkan transformasi citra serviks, hasil ekstraksi citra dan diagnosis kanker serviks menggunakan model RNN. Berdasarkan model RNN tebaik diperoleh nilai sensitivitas, spesifisitas, dan akurasi sebesar 97,78%, 95%, dan 92,3% untuk data *training* serta 87,5%, 100%, dan 76 % untuk data *testing*.

Kata Kunci : *graphical user interface* (GUI), *Gray Lavel Co-occurrence Matrix* (GLCM), kanker serviks, kolposkopi, *recurrent neural network* (RNN)

vii

metode skrining kanker leher rahim telah dikenal dan diaplikasikan, dimulai sejak tahun 1960-an dengan pemeriksaan tes Pap Smear. Selain itu dikembangkan metode visual dengan kolposkopi. Kolposkopi adalah suatu metode peneropongan leher rahim. Melalui kolposkopi diperoleh citra leher rahim (serviks) yang kemudian dilakukan pemeriksaan lebih lanjut untuk mendeteksi adanya sel yang abnormal pada serviks. Sebuah citra tersusun oleh kumpulan piksel-piksel yang membentuk suatu pola tertentu. *Artificial Neural Network* (ANN) merupakan salah satu metode yang dikenal baik untuk pengklasifikasian pola.

ANN merupakan suatu sistem yang memiliki karakteristik yang mirip dengan jaringan syaraf biologi (Fausett, 1994:3). ANN pertama kali diperkenalkan pada tahun 1943 oleh *neurophysiologist* Waren McCulloch dan *logician* Walter Pits. ANN merupakan jaringan yang meniru cara kerja sistem syaraf manusia (otak). Pemodelan ANN didasari oleh kemampuan otak manusia dalam mengorganisasi sel-sel penyusunnya (*neuron*), sehingga mampu melakukan pemrosesan informasi salah satunya pengklasifikasian pola-pola. Pada ANN dikenal algoritma *backpropagation*, algoritma ini sering digunakan dalam pemodelan ANN karena ke handalannya dalam menyelesaikan berbagai permasalahan terkait pengklasifikasian.

Berbagai penelitian terkait kanker serviks dan algoritma *backpropagation* pada ANN telah dilakukan sebelumnya antara lain, klasifikasi kanker serviks menggunakan citra koploskopi serviks dengan model *fuzzy* oleh Yushaila Nursajida (2013) menghasilkan tingkat keakurasian sebesar 40%. Agustina



PDF Complete

Your complimentary use period has ended.
Thank you for using PDF Complete.

Click Here to upgrade to Unlimited Pages and Expanded Features

Penelitian diagnosis kanker serviks dengan model RNN biasanya menggunakan jaringan *feedforward* serta dapat pula dikembangkan dengan jaringan *recurrent*. *Recurrent Neural Network* (RNN) adalah jaringan yang mengakomodasi *output* jaringan untuk menjadi *input* kembali (*recurrent*) pada jaringan dalam rangka menghasilkan *output* jaringan berikutnya. Beberapa penelitian yang menggunakan RNN antara lain, evaluasi kelayakan mahasiswa magang menggunakan RNN jaringan elman oleh Agus Aan Jiwa Permana dkk (2014) dan klasifikasi kanker kolon menggunakan *Recurrent Neural Network* oleh Reni Setyaningrum (2014) menghasilkan tingkat keakurasian 90%.

Perhitungan yang rumit dan kompleks pada RNN dapat dibantu dengan *software* Matrix Laboratory (MATLAB). Menurut Biran & Pärt-Enander (1999) MATLAB sangat cocok untuk RNN. MATLAB dapat membantu dalam merepresentasikan objek dari RNN, algoritma pembelajaran dan memvisualisasikan RNN dalam lingkungan simulink. Salah satu fasilitas visualisasi yang ada pada MATLAB adalah GUIDE (*Graphical User Interface (GUI) Builder*). Beberapa penelitian yang menggunakan GUI MATLAB antara lain Ronald Wilson (2008) merancang suatu program komposer musik berbasis GUI, serta aplikasi filter finite impulse response (FIR) untuk menghilangkan noise pada suara manusia yang dilakukan oleh Yeffry Handoko Putra dkk (2010).

C. Tujuan Penelitian

Tujuan penulisan penelitian ini menurut rumusan masalah di atas adalah:

1. Menjelaskan aplikasi model *Recurrent Neural Network* untuk diagnosis kanker serviks berdasarkan citra kolposkopi serviks.
2. Menjelaskan proses pembangunan diagnosis kanker serviks menggunakan model *Recurrent Neural Network* berbasis GUI MATLAB berdasarkan citra kolposkopi serviks
3. Mendeskripsikan tingkat ketepatan hasil diagnosis kanker serviks dengan model *Recurrent Neural Network* berdasarkan citra kolposkopi serviks.

D. Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini adalah:

1. Bagi penulis
Bagi penulis sendiri, penulisan skripsi ini dapat menambah pengetahuan dan wawasan tentang aplikasi model *Recurrent Neural Network* dan pemrograman GUI MATLAB dalam kehidupan sehari-hari, khususnya dalam bidang kesehatan.
2. Bagi para pembaca
Sebagai salah satu bahan dalam mempelajari model *Recurrent Neural Network* dan GUI MATLAB serta diharapkan penelitian ini dapat dijadikan sebagai referensi untuk penelitian selanjutnya. Selain itu, Penelitian ini diharapkan dapat membantu mempermudah dan mempercepat dalam mendiagnosis kanker serviks yang di derita pasien.

Melalui bantuan *software* MATLAB, dapat disusun berbagai program yang tampilannya dapat diperindah dengan bantuan GUI

Berdasarkan penelusuran terhadap penelitian terdahulu mengenai kanker serviks, RNN dan GUI MATLAB, penulis hendak melakukan kajian yang berjudul "Diagnosis Kanker Serviks Menggunakan Model *Recurrent Neural Network* (RNN) Berbasis *Graphical User Interface* (GUI)". Dimana akan dibangun sebuah program untuk mendiagnosis kanker serviks dengan menggunakan RNN dan tampilannya diperindah dengan GUI, harapannya dapat memberikan alternatif diagnosa dan mempermudah penggunaan program diagnosis ini.

B. Rumusan Masalah

Berdasarkan latar belakang di atas maka rumusan masalah yang akan dibahas adalah sebagai berikut:

1. Bagaimana mengaplikasikan model *Recurrent Neural Network* untuk diagnosis kanker serviks berdasarkan citra kolposkopi serviks?
2. Bagaimana proses pembangunan diagnosis kanker serviks menggunakan model *Recurrent Neural Network* berbasis GUI MATLAB berdasarkan citra kolposkopi serviks?
3. Bagaimana tingkat ketepatan model *Recurrent Neural Network* dalam mendiagnosis kanker serviks berdasarkan citra kolposkopi serviks ?

3. Bagi perpustakaan Universitas Negeri Yogyakarta

Penulisan skripsi ini juga bermanfaat dalam menambah koleksi bahan pustaka yang bermanfaat bagi Universitas Negeri Yogyakarta pada umumnya, dan mahasiswa Fakultas Matematika dan Ilmu Pengetahuan Alam pada khususnya.

leher rahim (serviks), suatu daerah pada organ reproduksi wanita yang merupakan pintu masuk ke arah rahim terletak antara uterus dan vagina. Kanker serviks merupakan kanker kedua yang paling umum diderita perempuan di seluruh dunia setelah kanker payudara (Deri Edianto, 2006:442) .

1. Penyebab

Penyebab terjadinya kanker serviks belum diketahui secara pasti, tetapi terdapat beberapa faktor resiko yang dapat berpengaruh terhadap terjadinya kanker serviks, sebagai berikut (Aden Rangiansanka, 2010:75):

a. HPV (*Human Papillomavirus*)

HPV merupakan penyebab utama kanker serviks. HPV merupakan sekelompok lebih dari 100 virus yang berhubungan yang dapat menginfeksi sel-sel pada permukaan kulit, ditularkan melalui kontak kulit seperti vagina, anal, atau *oral seks*. Beberapa tipe virus HPV yang beresiko tinggi menyebabkan kanker serviks adalah HPV tipe 16, 18, 31, 33 dan 45, virus ini dapat mengubah permukaan sel-sel vagina. Bila tidak segera terdeteksi dan diobati, infeksi HPV ini dalam jangka panjang dapat menyebabkan terbentuknya sel-sel pra- kanker serviks.

f. Pemakaian pil KB

Penggunaan pil KB jangka panjang dapat meningkatkan resiko terjadinya kanker serviks. Penelitian menunjukkan resiko kanker serviks menurun seiring penghentian penggunaan pil KB.

g. Infeksi Herpes Genitalis atau infeksi Klamidia menahun

Herpes genitalis dan klamidia adalah bakteri yang umum menyerang organ wanita, dan penularannya melalui hubungan seksual. Beberapa riset menemukan bahwa wanita yang memiliki sejarah atau infeksi bakteri ini beresiko terjangkit kanker serviks lebih tinggi dari pada wanita yang tidak menderita infeksi bakteri.

2. Gejala Kanker Serviks

Kanker serviks stadium awal umumnya tidak menimbulkan gejala yang khusus. Gejala kanker serviks biasanya baru muncul ketika kanker serviks telah menginjak stadium lanjut (Stadium IIB, III, dan IV). Berikut gejala-gejala kanker serviks (Aden Rangiansanka, 2010:77) :

- Pendarahan vagina yang abnormal, setelah 2 kali menstruasi, setelah melakukan hubungan seksual dan setelah menopause.
- Menstruasi tidak teratur .
- Keputihan yang menetap, dengan cairan yang encer , berwarna pink, coklat dan mengandung darah atau hitam serta berbau busuk.
- Nafsu makan berkurang, penurunan berat badan, kelelahan.
- Nyeri panggul, punggung atau tungkai.
- Dari vagina keluar air kemih atau tinja.
- Patah tulang (fraktur).

b. Merokok

Wanita yang merokok beresiko dua kali lebih mungkin menderita kanker serviks dibanding wanita yang tidak merokok. Rokok mengandung bahan karsinogenik yang dapat memicu kanker. Penelitian menunjukkan lendir serviks pada wanita perokok mengandung nikotin dan zat-zat lainnya yang ada di dalam rokok. Zat-zat tersebut akan menurunkan daya tahan serviks di samping merupakan kokarsinogen infeksi virus.

c. Hubungan seksual usia dini dan bergonta-ganti pasangan

Bergonta-ganti pasangan dapat memicu penularan berbagai penyakit kelamin, seperti *gardenella vaginosis* (keputihan berwarna abu-abu dan berbau), kalmidia, herpes dan penularan virus HPV yang dapat menyebabkan kanker serviks (Rama Diananda, 2007: 49).

d. Pemakaian DES (*Diethylstilbestrol*)

DES merupakan hormon yang digunakan wanita untuk mencegah keguguran. Obat ini digunakan antara tahun 1940-1971.

e. Gangguan sistem imun

Salah satu penyakit yang menyerang sistem imun atau kekebalan tubuh adalah HIV. HIV (*Human Immunodeficiency Virus*) adalah virus yang menyebabkan penyakit AIDS. HIV membuat sistem kekebalan tubuh seseorang berkurang sehingga tidak dapat memerangi baik infeksi HPV yang merupakan virus penyebab kanker serviks.

3. Deteksi

a. Tes Pap Smear

Secara umum, kanker serviks dapat dideteksi dengan mengetahui adanya perubahan pada daerah serviks dengan cara pemeriksaan sitologi menggunakan tes Pap Smear. Pap smear diperkenalkan oleh Dr. George Papanicolaou pada 1962 di Yunani. Melalui tes pap smear dapat diketahui apakah terdapat infeksi, radang, atau pertumbuhan sel-sel yang abnormal didalam serviks. *American Cancer Society* (ACS) dan *US Preventive Task Force* (USPTF) mengeluarkan panduan bahwa setiap wanita seharusnya melakukan tes pap smear 3 tahun sekali setelah pertama kali memulai aktivitas seksual atau saat berusia 21 tahun.

b. Tes IVA

Inspeksi Visual dengan Asam Asetat (IVA) adalah pemeriksaan leher rahim secara visual menggunakan asam cuka (asam asetat 3-5%) dengan mata telanjang untuk mendeteksi abnormalitas serviks (Depkes RI, 2009). IVA positif terdapat sel abnormal, jika ditemukan adanya area berwarna putih di sekitar zona transformasi. Berdasarkan penelitian yang dilakukan oleh Sankaranarayanan *et al* tentang perbandingan pasien kanker leher rahim yang meninggal dunia pada kelompok yang melakukan deteksi dini dengan IVA dan pada kelompok yang tidak melakukan deteksi dini pada negara berkembang (India) di dapatkan hasil bahwa mereka yang melakukan skrining IVA, 35% lebih sedikit yang meninggal dunia dibanding mereka yang tidak mendapat skrining IVA.

Owen Hans Timmerman pada 1925. Salah satu hasil pemeriksaan dengan kolposkopi adalah citra foto serviks. Pemeriksaan kolposkopi dilakukan guna konfirmasi apabila hasil tes *pap smear* menunjukkan adanya sel abnormal serta sebagai penentu biopsi.

d. Biopsi

Apabila hasil tes *pap smear* yang telah dikonfirmasi dengan pemeriksaan kolposkopi menunjukkan adanya sel abnormal dan lesi maka tahapan selanjutnya adalah biopsi. Biopsi adalah pengambilan sedikit jaringan serviks untuk diteliti oleh ahli patologi. Biopsi dilakukan didaerah yang abnormal jika sambungan skuamosa-kolumnar (SKK) yang terlihat seluruhnya oleh pemeriksaan kolposkopi.

4. Stadium Kanker Serviks

Kanker serviks berkembang secara bertahap, tetapi progresif. Proses terjadinya kanker serviks dimulai dengan mutasi sel-sel normal menjadi sel abnormal kemudian berkembang menjadi sel displastik sehingga terjadi kelainan epitel (displasia). Dari displasia kemudian berkembang menjadi karsinoma in situ yakni keberadaan sel-sel kanker masih terbatas hanya pada jaringan epitel.

Penentuan stadium kanker serviks menurut FIGO (*Federasi Internasional Ginekologi dan Obstetrik*) sebagai berikut :

a. Pembedahan

Pembedahan adalah terapi untuk membuang sel kanker dengan bantuan pisau bedah. Kanker bisa saja kembali muncul, oleh sebab itu setelah pembedahan dianjurkan untuk menjalani pemeriksaan ulang dan pap smear secara rutin.

b. Terapi penyinaran (radioterapi)

Radioterapi adalah terapi untuk menghancurkan kanker dengan sinar ionisasi. Kerusakan yang terjadi akibat radioterapi tidak hanya pada sel-sel kanker melainkan juga pada sel-sel normal disekitarnya, tetapi kerusakan pada sel kanker umunya lebih besar dari sel normal. Efek samping dari terapi penyinaran adalah sebagai berikut :

- iritasi rektum dan vagina,
- kerusakan kandung kemih dan rektum dan
- ovarium berhenti berfungsi.

c. Kemoterapi

Jika kanker sudah menyebar ke luar pangul dianjurkan untuk menjalani kemoterapi (Aden Ranggiainsanka, 2010:82). Kemoterapi adalah terapi untuk membunuh sel-sel kanker dengan menggunakan obat-obat antikanker yang disebut sitostatika. Terapi ini diberikan dalam suatu siklus, artinya suatu periode pengobatan diselingi dengan periode pemulihan, lalu dilakuakn pengobatan, diselingi dengan pemulihan dan begitu seterusnya.

Tabel 1. 1 Penentuan Stadium Kanker Serviks Menurut FIGO

Stadium 0	Karsinoma in situ, karsinoma intra epitel
Stadium I	Karsinoma masih terbatas di serviks
Stadium Ia	Invasi kanker ke stroma hanya dapat dikenali secara mikroskopik, lesi yang dapat dilihat secara langsung walau invasi yang sangat uperfisial dikelompokkan sebagai stadium Ib. Kedalaman infasi ke stroma tidak lebih dari 5 mm dan lebarnya lesi tidak lebih 7 mm.
Stadium Ia1	Invasi ke stroma dengan kedalaman tidak lebih dari 3 mm dan lebar tidak lebih dari 7 mm
Stadium Ia2	Invasi ke stroma dengan kedalaman tidak lebih dari 3 mm dan lebar tidak lebih dari 7 mm
Stadium Ib	Lesi terbatas di serviks atau secara mikroskopik lebih dari Ia
Stadium Ib1	Besar lesi secara klinis tidak lebih dari 4 cm
Stadium Ib2	Besar lesi secara klinis tidak lebih dari 4 cm
Stadium II	Telah melibatkan vagina tetapi belum sampai 1/3 bawah atau infiltrasi ke parametrium belum mencapai dinding panggul
Stadium Ila	Telah melibatkan vagina tapi belum melibatkan parametrium
Stadium IIb	Infiltrasi ke parametrium, tetapi belum mencapai dinding panggul
Stadium III	Telah melibatkan 1/3 bawah vagina atau adanya perluasan sampai dinding panggul. Kasus dengan hidronerosis atau gangguan fungsi ginjal dalam stadium ini, kecuali kelainan ginjal dapat dibuktikan oleh sebab lain.
Stadium IIIa	Keterlibatan 1/3 bawah vagina dan infiltrasi paramtrium belum mencapai dinding panggul.
Stadium IIIb	Perluasan sampai dinding panggul atau adanya hidronerosis atau gangguan fungsi ginjal.
Stadium IV	Perluasan ke luar organ reproduksi
Stadium IVa	Keterlibatan mukosa kandung kemih atau mukosa rectum
Stadium IVb	Metase jauh atau telah keluar dari rongga panggul

5. Pengobatan Kanker Serviks

Setelah kanker serviks didiagnosis, selanjutnya ditentukan pengobatan atau terapi yang tepat untuk penderita kanker. Pengobatan kanker serviks tergantung pada lokasi dan ukuran kanker, stadium kanker, usia, keadaan umum dan rencana penderita kanker untuk hamil lagi (Aden Ranggiainsanka, 2010:81).

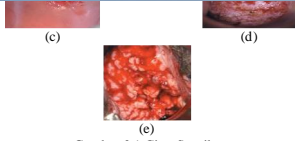
d. Imunoterapi

Imunoterapi adalah terapi untuk menguatkan daya tahan tubuh dan memperbesar kemampuan tubuh menghancurkan sel-sel kanker. Ada beberapa cara imunoterapi. Pertama, sel darah putih diambil dari sumsum tulang belakang kemudian dikembangkan dilaboratorium dan dimasukkan kemabali ketubuh. Cara kedua adalah dengan memberikan obat yang dapat merangsang sel-sel tubuh untuk memproduksi lebih banyak sel-sel darah putih untuk melawan sel kanker (Salinier *al*, 2011:29-30).

B. Tekstur

Tekstur merupakan pola berulang dari hubungan (distribusi) spasial dari derajat keabuan pada piksel - piksel yang berdekatan. Pola yang dimaksud adalah bentuk, kedalaman, warna, kecerahan dan sebagainya sedangkan piksel merupakan

Salah satu alat untuk mendeteksi keberadaan sel abnormal pada serviks adalah colposcope. Colposcope dilengkapi kamera yang dapat mendokumentasikan keadaan serviks serta area yang terindikasi terdapat sel abnormal. Melalui pemeriksaan menggunakan colposcope diperoleh citra kolposkopi serviks. Citra kolposkopi serviks dapat dikelompokkan menjadi lima kelas, yaitu citra serviks normal (a), stadium 1 (b), stadium 2 (c), stadium 3 (d), stadium 4 (e).



Gambar 2.1 Citra Serviks

normal (a), stadium 1 (b), stadium 2 (c), stadium 3 (d), dan stadium 4 (e) (Visual Inspection of the Cerviks Flash Card Set, Jhpiego)

Manusia melihat tekstur hanya berdasarkan deskripsi yang bersifat abstrak, seperti terang, gelap, halus, kasar, teratur, tidak teratur dan sebagainya. Penilaian seperti ini kurang tepat (non kuantitatif), oleh sebab itu diupayakan pendekatan kuantitatif atau matematis untuk memudahkan menganalisis tekstur. Salah satu teknik analisis tekstur adalah pengolahan citra digital (*Digital Image Processing*).

C. Pengolahan Citra Digital (*Digital Image Processing*)

Citra digital adalah sebuah gambar dalam tampilan dua dimensi yang disimpan dalam bentuk file sehingga dapat diolah menggunakan komputer (Gonzales&Woods, 2002:1). Citra digital terbagi menjadi empat jenis yakni citra RGB (bwarna), citra index, citra bersekala keabuan (*grayscale*) dan citra biner (McAndrew, 2004:12). Citra RGB adalah jenis citra yang menyajikan warna dalam bentuk komponen R (merah), G (hijau), dan B (biru). Kebanyakan citra

15

dengan,

$p(i,j)$ = piksel di lokasi baris ke- i dan kolom ke- j . $i = 1,2, \dots, N_g; j = 1,2, \dots, N_g$

N_g = banyak level abu-abu yang diperoleh dari citra.

2. Contrast

Contrast adalah ukuran intensitas kontras antara suatu piksel dan piksel lainnya dari keseluruhan citra (Sharma & Mukharjee, 2013:331). Rumus *contrast* (C) adalah sebagai berikut (Haralick, *et al.*, 1973:619):

$$C = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) (i-j)^2 \quad (2.2)$$

dengan,

$p(i,j)$ = piksel di lokasi baris ke- i dan kolom ke- j . $i = 1,2, \dots, N_g; j = 1,2, \dots, N_g$

N_g = banyak level abu-abu yang diperoleh dari citra.

3. Correlation

Correlation mengukur ketergantungan linear level abu-abu (*grey level*) antara piksel pada suatu posisi terhadap posisi lainnya dan dirumuskan sebagai berikut (Haralick, 1973:619):

$$Cr = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \frac{p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}}{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j)} \quad (2.3)$$

dengan,

$p(i,j)$ = piksel di lokasi baris ke- i dan kolom ke- j . $i = 1,2, \dots, N_g; j = 1,2, \dots, N_g$

17

bewarna hanya menyimpan sebagian kecil dari 16 juta warna yang mungkin, citra warna bertipe index merupakan citra yang mempunyai peta penyimpanan warna terkait index warna, yang hanya menyimpan daftar semua warna yang digunakan pada suatu citra. Citra *grayscale* adalah jenis citra yang menyajikan gradasi warna hitam dan putih sedangkan citra biner adalah citra dengan setiap pikselnya hanya dinyatakan dengan sebuah nilai dari dua kemungkinan, yaitu 0 menyatakan hitam dan 1 menyatakan putih. Salah satu teknik ekstraksi citra adalah *Gray Level Cooccurrence Matrix* (GLCM).

GLCM pertama kali diperkenalkan oleh Haralick di tahun 1973. Fitur-fitur yang diperoleh dari GLCM dapat membantu memahami rincian gambar secara keseluruhan (Gadkari, 2000:2). Pada analisis tekstur dengan metode GLCM digunakan *Matriks Intensitas Co-occurrence p(i,j)* adalah suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel. Fitur-fitur yang diperoleh dari ekstraksi GLCM antar lain *energy, contrast, correlation, sum of square, IDM, homogeneity, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy* (Haralick *et al.*, 1973), *dissimilarity, maximum probability* (Clausi, 2002).

1. Energy

Energy sering disebut *angular second moment* mengukur tingkat keseragaman nilai pixel, jika nilainya 1 berarti merupakan citra konstan (Sharma & Mukharjee, 2013:331). Rumus *energy* (E) adalah sebagai berikut (Haralick, *et al.*, 1973:619):

$$E = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j)^2 \quad (2.1)$$

16

N_g = banyak level abu-abu yang diperoleh dari citra.

$$\mu_x = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} i p(i,j),$$

$$\mu_y = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} j p(i,j),$$

$$\sigma_x = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu_x)^2 p(i,j)$$

$$\sigma_y = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (j - \mu_y)^2 p(i,j).$$

4. Sum of square variance

Menunjukkan variasi elemen-elemen matriks *co-occurrence*. Rumus *sum of square variance* (SSV) adalah sebagai berikut (Haralick *et al.*, 1973: 619):

$$SSV = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) (i - \mu)^2 \quad (2.4)$$

dengan,

$p(i,j)$ = piksel di lokasi baris ke- i dan kolom ke- j . $i = 1,2, \dots, N_g; j = 1,2, \dots, N_g$

N_g = banyak level abu-abu yang diperoleh dari citra.

$$\mu = \text{rata-rata} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j).$$

5. Inverse Difference Moment (IDM)

IDM adalah ukuran dari homogenitas lokal. Nilai IDM tinggi ketika level abu-abu (*grey level*) lokal seragam dan invers dari GLCM tinggi. (Mohanaiah, *et al.*, 2013:2). Rumus IDM adalah sebagai berikut (Haralick, *et al.*, 1973:619):

$$IDM = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{p(i,j)}{1 + |i-j|} \quad (2.5)$$

dengan,

18

Sum average menunjukkan jumlah dari rata-rata intensitas keabuan pada

citra. Rumus *sum average* (SA) adalah sebagai berikut (Harralick, et al, 1973:619):

$$SA = - \sum_{k=2}^{2N_g} \{ k (p_{x+y(k)}) \} \quad (2.6)$$

dengan,

$$p_{x+y(k)} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{i,j} ; i+j = k; k = 2,3, \dots, 2N_g,$$

N_g = banyak level abu-abu yang diperoleh dari citra.

7. Sum entropy

Sum entropy menunjukkan seberapa ketidakberaturan intensitas citra.

Rumus *sum entropy* (SE) adalah sebagai berikut (Harralick, et al, 1973:619):

$$SE = - \sum_{k=2}^{2N_g} p_{x+y(k)} \log(p_{x+y(k)}) \quad (2.7)$$

dengan,

$$p_{x+y(k)} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{i,j} ; i+j = k; k = 2,3, \dots, 2N_g,$$

N_g = banyak level abu-abu yang diperoleh dari citra.

8. Sum variance

Sum variance menunjukkan seberapa banyak level keabu-abuan yang bervariasi dari nilai rata-rata. Rumus *Sum variance* (SV) adalah sebagai berikut (Harralick et al, 1973:619):

19

N_g = banyak level abu-abu yang diperoleh dari citra.

11. Difference entropy

Difference entropy (DE) mengukur selisih entropy antara piksel yang berdekatan. Rumus *difference entropy* adalah sebagai berikut (Harralick, et al, 1973:619):

$$DE = - \sum_{k=0}^{N_g-1} (p_{x-y(k)}) \{ \log(p_{x-y(k)}) \} \quad (2.11)$$

dengan,

$$p_{x-y(k)} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{i,j} ; i-j = k; k = 0,1,2, \dots, (N_g - 1),$$

N_g = banyak level abu-abu yang diperoleh dari citra.

12. Maximum probability

Maximum probability (MP) menunjukkan tingkat abu-abu (grey level) yang memenuhi relasi pada persamaan entropi dan dirumuskan sebagai berikut (Anami & Burkpalli, 2009:11):

$$MP = \max_{i,j} \{ p(i,j) \} \quad (2.12)$$

dengan,

$p(i,j)$ = piksel di lokasi baris ke- i dan kolom ke- j .

13. Homogeneity

Homogeneity (H) menunjukkan keseragaman variasi intensitas dalam citra.

Secara matematis *Homogeneity* merupakan kebalikan dari *Contrast*. *Homogeneity* dirumuskan sebagai berikut (Sharma & Mukharjee, 2013:331):

$$H = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{p(i,j)}{1+|i-j|} \quad (2.13)$$

dengan,

21

$$SV = \sum_{k=2}^{2N_g} i - SE \sum_{k=2}^{2N_g} p_{x+y(k)} \quad (2.8)$$

dengan,

SE = jumlah entropi,

$$p_{x+y(k)} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{i,j} ; i+j = k; k = 2,3, \dots, 2N_g,$$

N_g = banyak level abu-abu yang diperoleh dari citra.

9. Entropy

Entropy merupakan ukuran keacakan dari intensitas citra. Nilai *entropy* akan besar untuk citra dengan transisi derajat keabuan merata dan bernilai kecil jika struktur citra tidak teratur. Rumus *entropy* (EN) dari suatu citra adalah sebagai berikut (Harralick et al, 1973:619):

$$EN = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{i,j} \log p_{i,j} \quad (2.9)$$

dengan,

$p(i,j)$ = piksel di lokasi baris ke- i dan kolom ke- j . $i =$

$1,2, \dots, N_g; j = 1,2, \dots, N_g$

N_g = banyak level abu-abu yang diperoleh dari citra.

10. Difference variance

Difference variance (DV) menunjukkan selisih varianse dua piksel yang berdekatan. Rumus *difference variance* adalah sebagai berikut (Harralick, et al, 1973:619):

$$DV = \text{varians dari } p_{x-y(k)} \quad (2.10)$$

dengan,

$$p_{x-y(k)} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{i,j} ; i-j = k; k = 0,1,2, \dots, (N_g - 1),$$

20

$p(i,j)$ = piksel di lokasi baris ke- i dan kolom ke- j . $i = 1,2, \dots, N_g; j =$

$1,2, \dots, N_g$

N_g = banyak level abu-abu yang diperoleh dari citra.

14. Dissimilarity

Dissimilarity (D) mengukur ketidakmiripan suatu tekstur atau perbedaan tiap piksel. *Dissimilarity* akan bernilai besar bila tekstur acak dan sebaliknya akan bernilai kecil bila seragam dan dapat dirumuskan sebagai berikut (Anami & Burkpalli, 2009:11):

$$D = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{i,j} |i-j| \quad (2.14)$$

dengan,

$p(i,j)$ = piksel di lokasi baris ke- i dan kolom ke- j . $i = 1,2, \dots, N_g; j =$

$1,2, \dots, N_g$

N_g = banyak level abu-abu yang diperoleh dari citra.

D. Artificial Neural Network (ANN)

Artificial Neural Network atau jaringan syaraf tiruan merupakan suatu sistem pemroses informasi yang memiliki karakteristik menyerupai jaringan syaraf biologis pada manusia (Siang, 2005:2). ANN merupakan generalisasi model matematika dari otak manusia yang meniru cara kerja sel otak manusia dalam menyelesaikan sebuah permasalahan. ANN diperkenalkan secara sederhana pada tahun 1943 oleh McCulloch dan Pitts. McCulloch dan Pitts melalui beberapa komputasi menggunakan neuron-neuron sederhana dan mengubahnya menjadi sistem neural yang mampu meningkatkan kemampuan komputasi. Model ANN

22

bagian yang lebih sederhana. Komponen terkecil dari sistem disebut neuron (unit), neuron adalah sebuah unit pemroses informasi yang merupakan dasar operasi pada ANN (Haykin, 1999:10). Neuron akan mentransformasikan informasi yang diterima menuju neuron lainnya, hubungan ini dikenal dengan nama bobot. Bobot akan menyimpan informasi pada suatu nilai tertentu. Bobot-bobot jaringan dapat berubah-ubah selama proses pelatihan jaringan bisa bertambah dan berkurang, perubahan bobot tersebut bertujuan untuk mendapatkan nilai *output* yang sesuai dengan *output* yang diharapkan. Terdapat 3 lapisan penyusun ANN yaitu lapisan *input* (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan *output* (*output layer*) (Siang, 2005: 24).

a. Lapisan *input*

Neuron pada lapisan input menerima masukan dari luar yang berupa gambaran dari suatu permasalahan.

b. Lapisan tersembunyi (*hidden layer*)

Lapisan tersembunyi tersusun atas neuron-neuron yang berperan meneruskan sinyal dari lapisan input. Neuron pada lapisan tersembunyi merupakan suatu sistem pemrosesan dimana terjadi pemrosesan informasi yang dapat meningkatkan kemampuan ANN dalam

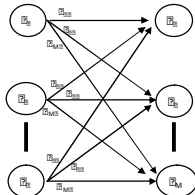
masing neuron. Arsitektur jaringan merupakan pola hubungan yang terjalin antar neuron. Sedangkan algoritma pelatihan merupakan metode untuk menentukan bobot-bobot pada jaringan.

1. Arsitektur Artificial Neural Network

Pola hubungan antar neuron pada ANN mengikuti pola tertentu tergantung pada arsitektur ANN. Pada dasarnya menurut Fausett (1994) terdapat 3 macam arsitektur :

a. Single-Layer Net (ANN dengan lapisan tunggal)

ANN dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini menerima *input* dari luar kemudian langsung mengolahnya menjadi *output* tanpa melalui lapisan tersembunyi. Jadi dapat dikatakan *single-layer net* hanya memiliki 1 lapisan *input* ($x_1, x_2, \dots, x_j, \dots, x_p, \quad j=1, \quad 2, \dots, p$) dan 1 lapisan *output* ($y_1, y_2, \dots, y_m, \dots, y_M, \quad m=1, \quad 2, \dots, M$) tanpa lapisan tersembunyi. Bobot yang menghubungkan lapisan *input* dan lapisan *output* adalah w_{mj} .

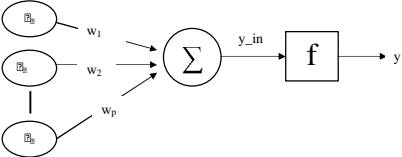


Gambar 2.3 ANN dengan Lapisan Tunggal (Fausett, 1994:13)

menyelesaikan masalah yang lebih kompleks. *Output* dari lapisan tersembunyi tidak dapat diamati secara langsung.

c. Lapisan *Output*

Neuron pada lapisan *output* disebut neuron *output*. Keluaran dari lapisan ini merupakan hasil ANN atau solusi dari permasalahan yang digambarkan pada lapisan *input*.



Gambar 2.2 ANN Sederhana (Sri Kusumadewi, 2010:72)

Pada Gambar 2.2 Sebuah neuron akan mengolah *input* ($x_1, x_2, \dots, x_j, \dots, x_p$) yang masing-masing memiliki bobot ($w_1, w_2, \dots, w_j, \dots, w_p$), dengan rumus:

$$y_{in} = \sum_{j=1}^p x_j w_j \tag{2.15}$$

Kemudian fungsi aktivasi f akan mengaktifkan y_{in} menjadi *output* jaringany. Keluaran dari lapisan *output* merupakan hasil ANN (solusi) dari suatu permasalahan yang diterima lapisan *input*. ANN memiliki kemampuan untuk mengenali dan membaca pola hubungan antara *input* dan *output* yang terdapat pada kumpulan data dari suatu permasalahan yang akan diselidiki.

Model ANN ditentukan oleh tiga hal (Fausett, 1994:3) yaitu, arsitektur jaringan, algoritma pelatihan dan fungsi aktivasi yang dijalankan di masing-

Sinyal *input* terbobot y_{in} ke neuron y_m , adalah sebagai berikut :

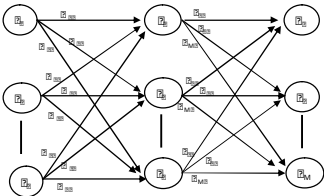
$$y_{in} = \sum_{m=1}^M v_{mj} x_j \tag{2.16}$$

Kemudian aktivasi y_{in} menggunakan suatu fungsi aktivasi f sehingga diperoleh y_m , sebagai berikut :

$$y_m = f(y_{in}) \tag{2.17}$$

b. Multilayer Net (ANN dengan banyak lapisan)

ANN dengan banyak lapisan memiliki satu atau lebih lapisan pada lapisan tersembunyi. Lapisan tersembunyi berada diantara lapisan *input* dan lapisan *output*. Terdapat sebuah lapisan bobot-bobot diantara dua lapisan yang bersebelahan. *Multilayer net* dapat menyelesaikan permasalahan yang lebih sulit, dimana *single-layer net* belum dapat menyelesaikannya dengan baik.



Gambar 2.4 ANN dengan Banyak Lapisan (Fausett, 1994:13)

Pada Gambar 2.4 ANN terdiri dari lapisan *input* ($x_1, x_2, \dots, x_j, \dots, x_p, \quad j=1, \quad 2, \dots, p$), lapisan tersembunyi ($z_1, z_2, \dots, z_k, \dots, z_K, \quad k=1, \quad 2, \dots, K$) dan lapisan *output* ($y_1, y_2, \dots, y_m, \dots, y_M, \quad m=1, \quad 2, \dots, M$). Bobot

diperoleh :

$$z_{in_k} = \sum_{j=1}^p x_j w_{kj}(x) \quad (2.18)$$

Nilai Z_k merupakan z_{in_k} yang telah diaktivasi menggunakan suatu fungsi aktivasi f , sehingga diperoleh :

$$z_k = f(z_{in_k}) \quad (2.19)$$

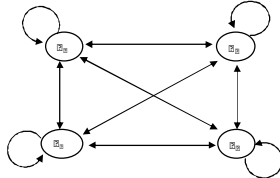
Kemudian sinyal *input* z_k digunakan untuk menghitung nilai y_{in_k}

$$y_{in_k} = \sum_{k=1}^K v_{mk} z_k \quad (2.20)$$

Nilai y_{in_k} kemudian di aktifasi menggunakan suatu fungsi aktivasi f , sehingga diperoleh :

$$y = f(y_{in_k}) \quad (2.21)$$

c. *Competitive Layer* (ANN dengan lapisan kompetitif)

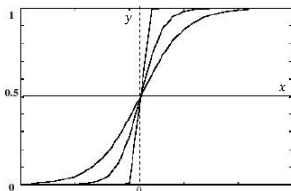


Gambar 2.5 ANN dengan Lapisan Kompetitif (Fausett, 1994:14)

ANN dengan lapisan kompetitif memiliki arsitektur jaringan yang berbeda dari jaringan dengan lapisan tunggal dan banyak lapisan. Pada

27

b. Fungsi Sigmoid Biner



Gambar 2.7 Fungsi Aktivasi Sigmoid Biner

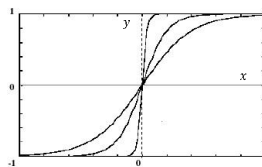
Fungsi sigmoid biner merupakan persamaan non-linear sehingga sangat baik untuk menyelesaikan permasalahan yang kompleks dan bersifat non-linear. Nilai *output* fungsi ini berkisar antara 0 sampai 1. Fungsi sigmoid biner dirumuskan sebagai berikut :

$$y = f \ x = \frac{1}{1 + e^{-x}} \quad (2.23)$$

dengan $f' \ x = \frac{e^{-x}}{1 + 2e^{-x} + e^{-2x}}$

Pada MATLABR2010a, sintaks untuk fungsi aktivasi sigmoid biner adalah *logsig*.

c. Fungsi Sigmoid Bipolar



Gambar 2.8 Fungsi Aktivasi Sigmoid Bipolar

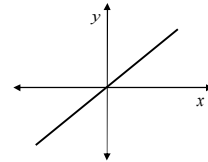
29

arsitektur ini antar neuron dapat saling terhubung. Salah satu contoh ANN dengan lapisan kompetitif adalah RNN (*Recurrent Neural Network*).

2. Fungsi Aktivasi

Fungsi aktivasi digunakan untuk mengaktifkan setiap neuron yang ada pada jaringan. Menurut Fausett (1994:17) fungsi aktivasi akan mengubah sinyal *input* menjadi sinyal *output* yang kemudian dikirim ke neuron lainnya. Pemilihan fungsi aktivasi tergantung kepada kebutuhan nilai *output* jaringan yang diinginkan (Anita Desiani & Muhammad Arhami, 2006:182). Beberapa fungsi aktivasi yang dapat digunakan sebagai pengaktif ANN, yaitu sebagai berikut (Fausett, 1994:17-19) :

a. Fungsi Identitas (Linear)



Gambar 2.6 Fungsi Identitas (Linear)

Fungsi identitas memiliki nilai *output* yang sama dengan nilai *input*. Fungsi identitas dirumuskan sebagai berikut :

$$y = f \ x = x \quad (2.22)$$

dengan $f' \ x = 1$

Pada MATLABR2010a, sintaks untuk fungsi identitas adalah *purelin*.

28

Fungsi sigmoid bipolar memiliki rentang nilai *output*nya berkisar antara -1 sampai 1. Fungsi sigmoid bipolar dirumuskan sebagai berikut :

$$y = f \ x = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (2.24)$$

dengan $f' \ x = \frac{2e^{-x}}{1 + 2e^{-x} + e^{-2x}}$

Pada MATLABR2010a, sintaks untuk fungsi sigmoid bipolar adalah *tansig* (Sri Kusumadewi, 2004:56-63).

3. Algoritma Pelatihan

Pelatihan merupakan bagian penting dari proses ANN karena tujuan utama dari pelatihan adalah untuk melakukan pengaturan bobot-bobot yang ada pada jaringan. Sehingga pada akhir pelatihan ANN diperoleh bobot-bobot akhir yang sesuai dengan pola data yang dilatihkan. Bobot-bobot tersebut nantinya dapat digunakan sebagai model ANN untuk memprediksi data *testing*. Selama proses pelatihan akan terjadi perbaikan bobot-bobot berdasarkan algoritma pelatihan tertentu. Algoritma pelatihan secara garis besar dibagi menjadi 2 yakni, pelatihan terawasi (*supervised learning*) dan pelatiantak terawasi (*unsupervised learning*) (Fausett, 1994:12-15).

a. Pelatihan terawasi (*supervised learning*)

Suatu algoritma pelatihan dikatakan terawasi jika *output* (target) yang diharapkan telah diketahuisebelumnya. Pada proses pembelajaran suatu pola input akan diberikan ke satu neuron pada lapisan *input*. Kemudian pola tersebut akan dirambatkan sampai ke neuron pada laisan *output*. Lapisan output akan mencocokkan nilai *output* yang dihasilkan dengan target yangdiharapkan. *Error*

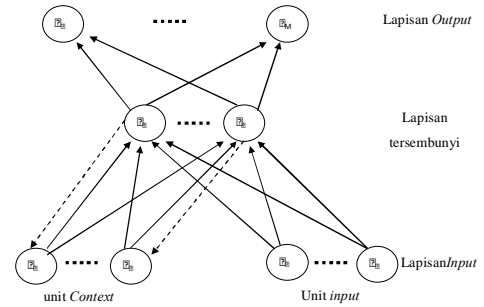
30

Pada algoritma pelatihan tak terawasi target *output* yang diinginkan tidak perlu diketahui sebelumnya. Pada algoritma ini, tidak dapat ditentukan hasil *output* yang diharapkan selama proses pembelajaran. Selama proses pelatihan, nilai bobot disusun dalam suatu range tertentu pada nilai *input* yang diberikan.

E. Recurrent Neural Network (RNN)

Artificial Neural Network umumnya menggunakan jaringan *feedforward* (perambatan maju) dalam pemodelan jaringannya. Salah satu tipe ANN yang dikembangkan dari dasar pemikiran pada jaringan *feedforward* adalah jaringan *recurrent* atau lebih dikenal sebagai *Recurrent Neural Network*. Jaringan *recurrent* lebih kompleks dibanding jaringan *feedforward* karena jika pada jaringan *feedforward* sinyal bergerak dari lapisan input kemudian melewati lapisan tersembunyi dan akhirnya mencapai neuron pada lapisan output dengan perambatan maju, hal berbeda terjadi pada jaringan *recurrent* dimana selain terjadi perambatan maju juga terjadi koneksi umpan balik (*feedback link*) dalam jaringan (Lin & Lee, 1996: 340). Suatu grup dari unit tertentu menerima koneksi umpan balik dari lapisan tersembunyi. Unit ini dikenal dengan nama *context unit* ($c_h, h=1,2,\dots,K$) seperti terlihat pada Gambar 2.9(Fausset, 1997:372).

31



Gambar 2.9 Arsitektur Simple Recurrent Neural Network (Fausset, 1994:373)

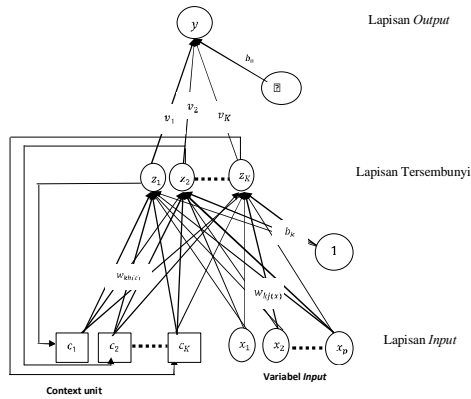
Jadi dapat dikatakan jaringan RNN adalah jaringan yang mengakomodasi *output* jaringan untuk menjadi *input* pada jaringan lagi yang kemudian digunakan untuk menghasilkan *output* yang baru. RNN merupakan jaringan dengan kemampuan dinamis karena perilaku jaringan tidak hanya bergantung pada *input* saat ini saja melainkan pada operasi sebelum jaringan. Terdapat 2 jenis jaringan RNN yaitu jaringan Hopfield dan jaringan Elman.

1. Jaringan Elman pada RNN

Jaringan Elman sering disebut *simple recurrent network* karena pada jaringan ini koneksi umpan balik (*feedback link*) hanya terdapat pada lapisan tersembunyi. Jaringan elman dibangun dengan 1 lapisan tersembunyi sedangkan jaringan Hopfield dibangun oleh lebih dari 1 lapisan tersembunyi dan mempunyai

32

koneksi umpan balik pada seluruh neuron pada setiap lapisannya. Arsitektur RNN jaringan Elman dapat dilihat pada Gambar 2.10.



Gambar 2.10 Arsitektur RNN Jaringan Elman

Pada Gambar 2.10 Jaringan terdiri dari lapisan *input* ($x_1, x_2, \dots, x_j, \dots, x_p, j = 1, 2, \dots, p$) lapisan tersembunyi ($z_1, z_2, \dots, z_K, k = 1, 2, \dots, K$) dan lapisan *output* (y). Bobot-bobot yang menghubungkan neuron *input* ke- j dan neuron lapisan tersembunyi ke- k adalah $w_{kj}(x)$. Sinyal-sinyal pada neuron *input* dirambatkan menuju neuron-neuron pada lapisan tersembunyi dengan bobot $w_{kj}(x)$ kemudian di bangkitkan dengan suatu fungsi aktivasi tertentu dan menghasilkan suatu *output* yakni c_h . Pada RNN jaringan Elman terjadi koneksi umpan balik (*feedback link*) antara lapisan tersembunyi dan lapisan *input* sehingga

33

c_h dijadikan *input* dalam jaringan kembali sebagai *context unit*. Bobot-bobot yang menghubungkan *context unit* ke- h dan lapisan tersembunyi ke- k adalah $w_{kh}(c)$. Pada RNN jaringan Elman lapisan tersembunyi terdiri dari 2 sinyal *input* terbobot yakni dari lapisan *input* dan *context unit*. Jumlah sinyal *input* terbobot tersebut kemudian diaktivasi dengan suatu fungsi aktivasi tertentu sehingga menghasilkan sinyal z_k . Sinyal z_k dirambatkan menuju neuron pada lapisan *output* y dengan bobot v_k , selanjutnya diperoleh sinyal terbobot yang di aktivasi dengan suatu fungsi aktivasi tertentu dan diperoleh nilai output y . Nilai *output* y merupakan hasil dari RNN dan merupakan solusi dari permasalahan yang didefinisikan di lapisan *input*. Adakalanya jaringan tidak mampu untuk mengakomodasi informasi yang adamelalui neuron *input* dan bobot-bobotnya. Guna menanggulangi hal tersebut, maka biasanya pada jaringan ditambah dengan bias (Sri Kusumadewi & Sri Hartati, 2010:73). Bobot bias yang terdapat pada lapisan tersembunyi dan lapisan *output* masing-masing adalah b_k dan b_o .

Model RNN pada jaringan Elman secara matematis dapat dirumuskan sebagai berikut (Boden, 2001:2) :

- Lapisan *input* → Lapisan tersembunyi (*hidden layer*)

$$z_k = f^*(z_{in\ k}) \quad (2.25)$$

$$z_{in\ k} = b_k + \sum_{j=1}^p x_j w_{kj} + \sum_{h=1}^K c_h w_{kh} \quad (2.26)$$

- Lapisan tersembunyi (*hidden layer*) → Lapisan *Output*

$$y = f^o(y_{in}) \quad (2.27)$$

$$y_{in} = \sum_{k=1}^K v_k z_k + b_o \quad (2.28)$$

34

y : nilai *output*
 f^o : fungsi aktivasi pada lapisan tersembunyi ke lapisan *output*
 f^* : fungsi aktivasi pada lapisan *input* ke lapisan tersembunyi
 v_k : bobot dari neuron ke- k pada lapisan tersembunyi yang menuju lapisan *output* dengan $k=1,2,...,K$,
 x_j : variabel *input* dengan $j=1,2,...,p$,
 $w_{kj\ x}$: bobot dari lapisan *input* ke- j menuju neuron ke- k pada lapisan tersembunyi,
 c_h : variabel *input* yang berada pada *konteks unit* dengan $h=1,2,...,K$,
 $w_{k(h)c}$: bobot *konteks unit*- h menuju neuron ke- k pada lapisan tersembunyi,
 b_k : bobot bias pada neuron ke- k pada lapisan tersembunyi dengan $k=1,2,...,K$
 b_0 : bobot bias pada neuron lapisan *output*
 ε : *error*

2. Algoritma Backpropagation pada RNN jaringan Elman

Algoritma *backpropagation* pertama kali dicetuskan oleh Paul Werbos pada tahun 1974. Pelatihan sebuah jaringan menggunakan algoritma *backpropagation* terdiri dari 3 fase yaitu perambatan maju (*feedforward*), perambatan mundur (*backward*) dan penyesuaian bobot. Pada fase *feedforward* neuron-neuron *input* dirambatkan dari lapisan *input* ke lapisan *output*

35

Perintah: net.trainParam.goal = TargetError

3) Learning rate

Learning rate adalah laju pelatihan. Semakin besar nilai *learning rate*, semakin besar pula langkah pelatihan. Semakin kecil *learning rate*, maka proses pelatihan akan sangat lama. Sehingga perlu pemilihan nilai yang tepat untuk *learning rate*. Nilai default untuk *learning rate* adalah 0.01

Perintah: net.trainParam.lr = LearningRate

4) Rasio untuk menaikkan learning rate

Rasio ini berguna sebagai faktor pengali untuk menaikkan *learning rate* apabila *learning rate* yang ada terlalu rendah untuk mencapai konvergensi.

Perintah: net.trainParam.lr.inc=IncLearningRate. Nilai default untuk rasio kenaikan *learning rate* adalah 1.05.

5) Rasio untuk menurunkan learning rate

Rasio ini berguna sebagai faktor pengali untuk menurunkan *learning rate* apabila *learning rate* yang ada terlalu tinggi untuk menuju ketidakstabilan. Nilai default untuk rasio penurunan *learning rate* adalah 0.7.

Perintah: net.trainParam.lr.dec=DecLearningRate

6) Jumlah epoch yang akan ditunjukkan kemajuannya

Menunjukkan jumlah epoch akan ditampilkan. Nilai default untuk jumlah epoch yang akan ditunjukkan adalah 25.

Perintah: net.trainParam.show = EpochShow

7) Maksimum kenaikan kinerja

37

menggunakan fungsi aktivasi tertentu untuk memperoleh *error*. *Error* merupakan selisih nilai *output* RNN dengan nilai *output* target sebenarnya. Algoritma *backpropagation* menggunakan *error* tersebut untuk penyesuaian bobot-bobot jaringan dalam arah mundur (*backward*). Fungsi aktivasi yang digunakan pada algoritma *backpropagation* harus memenuhi beberapa syarat yakni fungsi aktivasi tersebut kontinu, terdiferensial, dan monoton naik contohnya seperti *sigmoid biner*, *sigmoid bipolar* dan *linear* (Fausett, 1994:292-293).

Sesudah pelatihan, aplikasi dalam jaringan hanya terdiri dari fase *feedforward*.

Secara umum tahapan Algoritma *backpropagation* adalah sebagai berikut:

- Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil)
- Tetapkan : Maksimum epoch, target *error*, *Learning Rate* dan parameter lainnya, yaitu :

1) Maksimum epoch

Maksimum epoch adalah jumlah epoch maksimum yang boleh dilakukan selama proses pelatihan. Iterasi akan dihentikan apabila nilai epoch melebihi maksimum epoch. Nilai default untuk maksimum epoch adalah 10.

Perintah: net.trainParam.epochs = MaxEpoch

2) Kinerja tujuan

Kinerja tujuan adalah target nilai fungsi kinerja. Iterasi akan dihentikan apabila nilai fungsi kinerja kurang dan atau sama dengan kinerja tujuan. Nilai default untuk kinerja tujuan adalah 0.

36

Maksimum kenaikan kinerja adalah nilai maksimum kenaikan *error* yang diijinkan, antara *error* saat ini dan *error* sebelumnya. Nilai default untuk maksimum kenaikan kinerja adalah 1.04.

Perintah: net.trainParam.max_perf_inc=MaxPerfInc

8) Momentum

Guna mempercepat waktu pelatihan, prosedur perubahan bobot dimodifikasi dengan menggunakan momentum. Momentum adalah perubahan bobot yang didasarkan atas arah gradient pola terakhir dan pola sebelumnya. Besarnya momentum antara 0 sampai 1. Apabila nilai momentum = 0, maka perubahan bobot hanya akan dipengaruhi oleh gradiennya. Tetapi, apabila nilai momentum = 1, maka perubahan bobot akan sama dengan perubahan bobot sebelumnya.

Perintah: net.trainParam.mc = Momentum

- Kerjakan langkah-langkah berikut selama (Epoch < Maksimum Epoch) dan (MSE < Target Error):

Epoch = Epoch + 1

1. Fase I (*feedforward*)

- Tiap-tiap neuron *input* (x_j , $j=1,2,3,...,p$) menerima sinyal x_j dan meneruskan sinyal tersebut ke semua neuron pada lapisan tersembunyi (*hidden layer*).

- Tiap-tiap neuron pada lapisan tersembunyi (z_k , $k=1,2,...,K$) menjumlahkan sinyal-sinyal *input* terbobot sebagai berikut:

$$z_{in\ k} = b_k + \sum_{j=1}^p x_j w_{kj\ x} \quad (2.30)$$

38

untuk menghitung sinyal *output* digunakan fungsi aktivasi *sigmoid*

bipolar (2.24) sebagai berikut:

$$z(1)_k = f \quad z_{in \ 1 \ k} \quad (2.31)$$

$$z(1)_k = \frac{1 - e^{-z_{in \ 1 \ k}}}{1 + e^{-z_{in \ 1 \ k}}} \quad (2.32)$$

kemudian sinyal tersebut akan dikirim ke semua neuron di lapisan *output* dan *context unit* di lapisan *input*.

- c. *Context unit* (c_h , $h = 1, 2, 3 \dots K$) menerima dan mengirim sinyal ke lapisan tersembunyi. Sinyal yang diterima merupakan sinyal *output* lapisan tersembunyi yang telah diaktivasi (Boden, 2011:2).

$$c_h = z(1)_k \quad (2.33)$$

Setiap neuron *context unit* kemudian mengirimkan kembali sinyal tersebut ke lapisan tersembunyi sebagai berikut:

$$z_{in \ 2 \ k} = \sum_{h=1}^K c_h w_{kh \ c} \quad (2.34)$$

c_h : *context unit*, sinyal *output* yang telah diaktivasi, $h=1, 2, \dots, K$
 $w_{kh(c)}$: bobot dari *context unit* ke- h menuju neuron ke- k pada lapisan tersembunyi

- d. Setiap neuron pada lapisan tersembunyi z_k , ($k=1, 2, \dots, K$) menjumlahkan sinyal-sinyal *input* terbototyng diperoleh, yaitu:

39

kemudian sinyal tersebut dikirim ke neuron pada lapisan *output*. Fungsi aktivasi yang digunakan dari lapisan *input* ke lapisan tersembunyi dalam tugas akhir ini adalah *sigmoid bipolar* (2.24) , sehingga diperoleh :

$$z_k = \frac{1 - e^{-z_{in \ k}}}{1 + e^{-z_{in \ k}}} \quad (2.37)$$

- e. Neuron *output*(y) menjumlahkan sinyal-sinyal *input* terbotob dari lapisan tersembunyi menggunakan rumus berikut:

$$y_{in} = b_0 + \sum_{k=1}^K v_k z_k \quad (2.38)$$

dengan,

b_0 : bobot bias pada neuron *output*

v_k : bobot dari neuron ke- k pada lapisan tersembunyi yang menuju lapisan *output* dengan $k=1, 2, \dots, K$

z_k : sinyal *output* yang telah diaktivasi dari lapisan tersembunyi dengan $k=1, 2, \dots, K$

Untuk menghitung sinyal *output* y_{in} dihitung menggunakan fungsi aktivasi *linear* (2.22) sebagai berikut:

$$y = f \quad y_{in} \quad (2.39)$$

2. Fase II (Backpropagation)

- a. Neuron *output* (y) menerima target pola yang berhubungan dengan pola *input* pelatihan, kemudian dihitung informasi *error*nya:

$$\delta_0 = t - y \quad f' \quad y_{in} \quad (2.40)$$

- 1) Sinyal *output* yang diterima lapisan tersembunyi dari lapisan *input*:

$$z_{in}(1)_k = b_k + \sum_{j=1}^p x_j w_{kj(c)} \quad (2.30)$$

- 2) Sinyal *output* yang diterima lapisan tersembunyi dari *context unit*

$$z_{in}(2)_k = \sum_{h=1}^K c_h w_{kh(c)} \quad (2.34)$$

Dari persamaan (2.30) dan (2.34)diperoleh,

$$z_{in \ k} = z_{in \ 1 \ k} + z_{in \ 2 \ k} = b_k + \sum_{j=1}^p x_j w_{kj \ x} + \sum_{h=1}^K c_h w_{kh(c)} \quad (2.35)$$

dengan,

$z_{in}(1)_k$: sinyal yang diterima lapisan tersembunyi dari neuron lapisan *input*

$z_{in}(2)_k$: sinyal yang diterima lapisan tersembunyi dari *konteks unit*

b_k : bobot bias pada neuron ke- k pada lapisan tersembunyi dengan $k=1, 2, \dots, K$

x_j : variabel *input*, $j=1, 2, \dots, p$

c_h : *konteks unit* atau sinyal *output* (telah diaktivasi) yang diterimalapisan tersembunyi dari neuron *input*= $1, 2, \dots, K$

$w_{kj(x)}$: bobot dari lapisan *input* ke- j menuju neuron ke- k pada lapisan tersembunyi

$w_{kh(c)}$: bobot dari *konteks unit* ke- h menuju neuron ke- k pada lapisan tersembunyi

untuk menghitung sinyal *output* $z_{in \ k}$ digunakan fungsi aktivasi sebagai berikut:

$$z_k = f(z_{in \ k}) \quad (2.36)$$

40

t adalah target *output* yang diinginkan, kemudian dihitung koreksi bobot yang nantinya akan digunakan untuk memperbaiki nilai v_k dengan rumus sebagai berikut:

$$\Delta v_k = \alpha \delta_0 z_k \quad (2.41)$$

dimana α adalah *learning rate*, kemudian menghitung koreksi bias yang nantinya akan digunakan untuk memperbaiki nilai b_0 , dan kirimkan δ_0 ke neuron-neuron pada lapisan sebelumnya dengan rumus

$$\Delta b_0 = \alpha \delta_0 \quad (2.42)$$

- b. Tiap-tiap neuron tersembunyi (z_k , $k = 1, 2, \dots, K$) menjumlahkan delta *input*nya dari neuron-neuron yang berada pada lapisan di atasnya sebagai berikut:

$$\delta_{in \ k} = \sum_{k=1}^K \delta_0 v_k \quad (2.43)$$

untuk menghitung informasi *error*, kalikan nilai $\delta_{in \ k}$ dengan turunan dari fungsi aktivasinya:

$$\delta_k = \delta_{in \ k} f' \quad z_{in \ k} \quad (2.44)$$

kemudian dihitung koreksi bobot yang nantinya akan digunakan untuk memperbaiki nilai $w_{kj(x)}$ dan $w_{kh(c)}$ dengan rumus sebagai berikut:

$$\Delta w_{kj(x)} = \alpha \delta_k x_j \quad (2.45)$$

$$\Delta w_{kh(c)} = \alpha \delta_k c_h \quad (2.46)$$

selain itu, menghitung koreksi bias yang nantinya akan digunakan untuk memperbaiki nilai b_k :

$$\Delta b_k = \alpha \delta_k \quad (2.47)$$

- b. Tiap-tiap neuron lapisan tersembunyi ($z_k, k = 1, 2, \dots, K$) memperbaiki bias dan bobotnya sebagai berikut:

$$w_{kj(x)} \text{ baru} = w_{kj(x)} \text{ lama} + \Delta w_{kj} \quad (2.50)$$

$$w_{kh(c)} \text{ baru} = w_{kh(c)} \text{ lama} + \Delta w_{kh} \quad (2.51)$$

$$b_k \text{ baru} = b_k \text{ lama} + \Delta b_k \quad (2.52)$$

3. Membangun RNN Jaringan Elman Algoritma *Backpropagation* pada MATLAB2010a

Pada MATLAB2010a, untuk membangun RNN jaringan Elman dengan algoritma *Backpropagation* dapat menggunakan fungsi *newelm* :

$$net = newelm(PR, S1 \ S2 \dots SN1, TF_1 TF_2 \dots TF_{N1}, BTF_1, BLF_1, PF)$$

dengan,

PR = matriks berukuran $R \times 2$ yang berisi nilai minimum dan maksimum dari setiap variabel *input*, dengan R adalah jumlah variabel *input*.

S_i = jumlah neuron pada lapisan ke- i .

$N1$ = jumlah lapisan tersembunyi

TF_i = fungsi aktivasi pada lapisan ke- i (default : *tansig*)

BTF_i = fungsi pelatihan untuk algoritma *backpropagation* (default : *traingdx*)

BLF_i = pelatihan untuk perbaikan bobot-bobot pada *backpropagation* (default :

G. Prosedur Pemodelan RNN

Langkah-langkah pemodelan RNN jaringan Elman algoritma *Backpropagation* untuk diagnosis kanker serviks, sebagai berikut :

1. Mendefinisikan variabel *input* dan target

RNN hanya dapat mengolah data dalam bentuk numerik oleh sebab itu apabila data yang digunakan berupa citra (bukan data numerik) maka citra tersebut harus di ekstraksi terlebih dahulu guna mendapatkan nilai numerik dari citra. Data dari variabel *input* dan target digunakan selama pelatihan RNN untuk memperoleh bobot dan model terbaik serta untuk pengujian model RNN. Variabel *input* merupakan parameter-parameter yang mempengaruhi nilai *output* atau target. Sedangkan target merupakan keluaran (*output*) yang diharapkan dari hasil pelatihan RNN.

2. Pembagian data

Sebelum proses pelatihan RNN dilakukan data yang diperoleh dibagi menjadi dua bagian yakni data *training* dan data *testing*. Ukuran data *training* harus merupakan data yang paling besar dan digunakan dalam pelatihan guna mempelajari pola dalam data. Sedangkan ukuran data *testing* berkisar antara 10%-30% dari ukuran data *training*, data *testing* digunakan untuk pengujian model RNN dari hasil pelatihan (Kaastra, 1996: 223). Pembagian data yang dapat digunakan adalah 75% untuk data *training* dan 25% untuk data *testing* (Lin *et al*, 2008:1).

learnqdm)

PF = fungsi kinerja (default : *mse*)

F. Mean Square Error (MSE) dan Prinsip Parsimoni

MSE merupakan salah satu metode untuk mengevaluasi hasil peramalan atau prediksi dari suatu model. *Error* dari setiap data di kuadratkan dan dijumlahkan kemudian dibagi dengan banyaknya data pengamatan. Rumus untuk menghitung MSE adalah sebagai berikut (Hanke & Wichern, 2005:79-80):

$$MSE = \frac{1}{n} \sum_{i=1}^n e_i^2 \quad (2.53)$$

dengan,

MSE : *meansquareerror*

e_i : *error* ke- i (selisih target ke- i dan *output* ke- i)

n : banyaknya data

Prinsip parsimoni pertama kali diperkenalkan oleh seorang logikawan bernama William of Ockham pada abad ke-14. Suatu model matematis harus sesuai dengan kompleksitas permasalahan yang dihadapi hal ini guna menghindari terjadinya *overfitting*. Menurut prinsip parsimoni model dengan asumsi dan variabel yang sederhana tetapi memiliki kekuatan penjelasan terbesar harus dipilih sebagai model terbaik untuk menyelesaikan suatu masalah (McLeod, 1991:4). Penjelasan yang baik adalah penjelasan yang sederhana (Crawley, 2014:8).

3. Normalisasi (*Preprocessing*)

Sebelum melakukan proses pelatihan data perlu dinormalisasikan. Tujuan dari normalisasi data adalah menghomogenkan data sehingga memiliki *mean* = 0 dan standar *deviasi* = 1. Rumus normalisasi (Samarasinghe, 2007:253) adalah sebagai berikut:

$$x_i^* = \frac{x_i - \bar{x}}{s} \quad (2.54)$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.55)$$

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (2.56)$$

dengan,

x_i^* : nilai data ke- i yang telah dinormalisasi

x_i : nilai data ke- i

\bar{x} : rata-rata nilai data

s : nilai standar deviasi data

n : banyaknya data pengamatan

Normalisasi data dapat menggunakan bantuan MATLAB2010a dengan memanggil fungsi *prestd* (Sri Kusumadewi, 2004:191):

$$[Pn, meanp, stdp, tn, meant, stdt] = prestd(P, T)$$

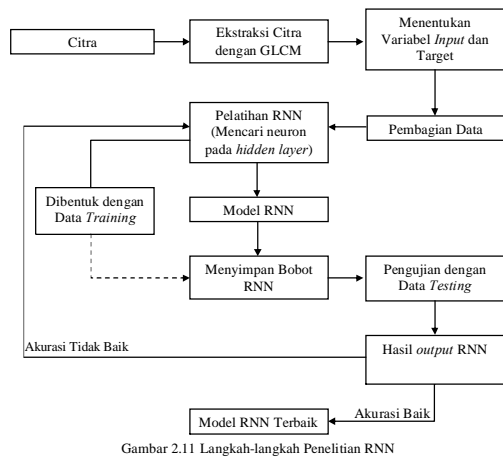
Apabila pada RNN yang telah melalui proses pelatihan menggunakan fungsi *prestd* untuk normalisasi, maka apabila ada data *input* baru atau data *testing* yang akan di simulasikan dengan jaringan, maka data *input* baru atau data *testing* tersebut perlu disesuaikan dengan *mean* dan standar *deviasi* dari jaringan. Pada MATLAB2010a dapat menggunakan fungsi *trastd* :

- T : Transpose dari matriks target
- Pn : Matriks *input* yang telah ternormalisasi
- Tn : Matriks target yang telah ternormalisasi
- Qn : Matriks data *input* baru yang telah ternormalisasi
- Meanp : *mean* pada matriks (P)
- Stdp : Standar *deviasi* pada matriks (P)
- Meant : *Mean* pada matriks (T)
- StdT : Standar *deviasi* pada matriks (T)

4. Perancangan model RNN

Perancangan model RNN terbaik dimulai dengan menentukan banyak neuron pada lapisan tersembunyi secara *trial and error*. Banyak neuron pada lapisan tersembunyi disesuaikan dengan kompleksitas permasalahan yang akan diselidiki (Sri Kusumadewi, 2004:327). Proses pelatihan RNN jaringan Elman dengan algoritma *Backpropagation* terdiri dari 3 fase yakni fase *feedforward*, fase *backpropagation* dan perubahan bobot. Selama proses pelatihan data yang digunakan adalah data yang telah dinormalisasi.Model terbaik RNN dipilih dengan membandingkan nilai MSE (2.53) data *training* dan *testing*. Prinsip parsimoni digunakan dalam membandingkan nilai MSE yang diperoleh dari hasil

47



Gambar 2.11 Langkah-langkah Penelitian RNN

H. Graphical User Interface Builder (GUIDE)

MATLAB adalah salah satu bahasa pemrograman yang biasa digunakan pada teknik-teknik komputasi. MATLAB menyediakan fasilitas-fasilitas untuk komputasi, visualisasi, dan pemrograman. Salah satu fasilitas yang tersedia di MATLAB adalah GUIDE atau *Graphical User Interface (GUI) builder*.GUIDE merupakan sebuah GUI yang dibangun dengan objek grafik seperti tombol (*button*), kotak teks, *slider*, *menu* dan lain-lain. Sedangkan GUI merupakan

49

pelatihan. Berdasarkan prinsip parsimoni, model yang terbaik adalah model yang mempunyai nilai MSE kecil dengan jaringan yang sederhana. Misalkan terdapat dua model yang memiliki nilai MSE yang hampir sama, maka model yang dipilih adalah arsitektur jaringan dengan jumlah neuron tersembunyi yang paling minimal.

5. Denormalisasi data(*postprocessing*)

Setelah dinormalisasi data perlu dikembalikan kekeadaan semula (denormalisasi). Pada MATLABR2010a perintah denormalisasi dapat dilakukan dengan memanggil fungsi *poststd* sebagai berikut :

```
[P,T]=poststd(Pn,meanp,stdp,Tn,meant,stdt)
```

Sebelum melakukan proses denormalisasi, terlebih dahulu dilakukan simulasi pada jaringan syaraf dengan perintah sebagai berikut (Sri Kusumadewi, 2004:192):

```
Misal: an = sim(net,Pn);
a = poststd(an,meant,stdt);
```

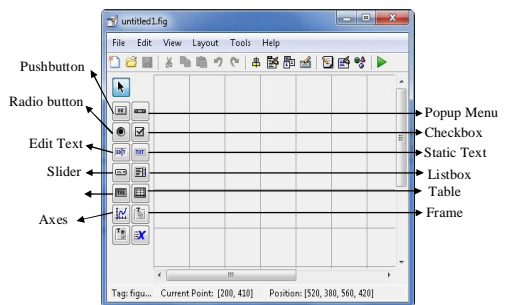
Secara umum prosedur pembentukan model RNN digambarkan pada gambar 2.11.

48

tampilan grafis hasil dari GUIDE yang bertujuan untuk memudahkan pengguna untuk berinteraksi. Program yang menggunakan GUI umumnya lebih mudah dipelajari dan digunakan karena orang yang tidak mengetahui pemrograman MATLAB sekalipun dapat menjalankan program tersebut.GUI berguna untuk menampilkan *software* yang dibuat (Wittman, 2008:2).Berdasarkan kelebihan-kelebihan GUIDE pada MATLAB, maka pada penelitian skripsi ini akan menggunakan GUI untuk menghasilkan tampilan yang lebih baik dan interaktif terhadap pengguna. Memulai GUIDE MATLAB dapat dilakukan dalam dua cara yakni :

- Melalui command MATLAB dengan mengetik : guide.
- Klik tombol Start MATLAB dan pilih MATLAB, lalu pilih GUIDE (GUI Bulder).

Penyimpanan akhir GUI terdiri dari dua ekstensi yakni file *fig-file* dan *m-file*.Berikut adalah tampilan GUI dengan perintah guide



Gambar 2.12 Komponen UIcontrol

Pushbutton merupakan jenis kontrol berupa tombol tekan yang akan menghasilkan tindakan jika diklik, misalnya tombol OK, Keluar, Save, Proses, dan sebagainya.

2. Toggle Button

Toggle button memiliki fungsi yang sama dengan pushbutton. Perbedaan adalah saat pushbutton ditekan, maka tombol akan kembali pada posisi semula jika tombol mouse dilepas, sedangkan pada toggle button, tombol tidak akan kembali ke posisi semula, kecuali toggle button ditekan kembali.

3. Radio Button

Radio button digunakan untuk memilih atau menandai satu pilihan dari beberapa pilihan yang ada.

4. Edit Text dan Static Text

Edit text digunakan untuk memasukkan atau memodifikasi suatu text yang diinputkan dari keyboard, sedangkan static text hanya berguna untuk menampilkan text/tulisan, sehingga tidak bisa dimodifikasi atau diedit.

5. Frames& Tabel

Frames merupakan kotak tertutup yang dapat digunakan untuk mengelompokkan kontrol-kontrol yang berhubungan. Tabel berguna untuk

spesifisitas dan akurasi. Rumus untuk menghitung nilai sentivitas dan spesifisitas (Simundic, 2009:3) adalah sebagai berikut:

Tabel 2.1 Uji Sensitivitas dan Spesifisitas dalam Tabel 2 x 2

Tes	Kanker		Total
	Ada	Tidak Ada	
Positive	True Positive (TP)	False Positive (FP)	TP + FP
Negative	False Negative (FN)	True Negative (TN)	FN + TN
Total	TP + FN	FP + TN	TP + FP + FN + TN

TP : pasien memiliki kanker dan hasil diagnosa menyatakan pasien memiliki kanker

TN : pasien tidak memiliki kanker dan hasil diagnosa menyatakan pasien tidak memiliki kanker

FP : pasien tidak memiliki kanker dan hasil diagnosa menyatakan pasien tidak memiliki kanker

FN : pasien memiliki kanker dan hasil diagnosa menyatakan pasien tidak memiliki kanker

Sensitivitas adalah kemampuan model diagnosis untuk mengidentifikasi dengan benar pasien yang memiliki kanker dari semua pasien yang memiliki kanker (Kanchanaraksa, 2008: 16). Misalnya diperoleh nilai sensitifitas 95%, artinya jika ada pasien yang memang menderita kanker serviks melakukan uji diagnosa, pasien tersebut dinyatakan 95% menderita kanker serviks. Rumus sensitifitas adalah sebagai berikut (Simundic, 2009: 4) :

$$Sensitivitas = \frac{TP}{TP+FN} \times 100\% \quad (2.57)$$

menampilkan nilai, penggunaan tabel hampir sama seperti kolom-kolom pada ms excel.

6. Checkboxes

Kontrol checkboxes digunakan menyediakan beberapa pilihan mandiri atau tidak bergantung dengan pilihan-pilihan lainnya. Contoh aplikasi penggunaan checkboxes adalah ketika kita diminta untuk memilih hobi. Karena hobi bisa lebih dari satu, maka kita dapat mengklik checkboxes lebih dari satu kali.

7. Slider

Slider berguna untuk menginput nilai tanpa menggunakan keyboard, tatapi hanya dengan cara menggeser slider secara vertical maupun horizontal ke nilai yang kita inginkan.

8. Popup Menu

Popup menu berguna menampilkan daftar pilihan atau menu yang didefinisikan pada *String Property*. Ketika tidak dibuka, popup menu hanya menampilkan satu item yang menjadi pilihan pertama pada String Property.

9. Axes

Axes berguna untuk menampilkan sebuah grafik atau gambar (image). Axes sebenarnya tidak masuk dalam *UIControl*, tetapi axes dapat diprogram agar pemakai dapat berinteraksi dengan axes dan obyek grafik yang ditampilkan melalui axes.

1. Uji Ketepatan Diagnosis

Suatu model diagnosis dapat memberikan hasil diagnosis yang tidak tepat. Tingkat keepatan suatu model diagnosis dapat dilihat dari nilai sensitivitas,

Spesifisitas adalah kemampuan model diagnosis untuk mengidentifikasi dengan benar pasien yang tidak memiliki kanker dari semua pasien yang tidak memiliki kanker (Kanchanaraksa, 2008: 15).Misalnya diperoleh nilai spesifisitas 95%, artinya jika ada pasien yang memang tidak menderita kanker serviks melakukan uji diagnosa , pasien tersebut dinyatakan 95% tidak menderita kanker serviks. Rumus spesifisitas adalah sebagai berikut (Simundic, 2009: 3):

$$Spesifisitas = \frac{TN}{TN+FP} \times 100\% \quad (2.58)$$

Akurasi adalah kemampuan suatu model diagnosis untuk mengidentifikasi dengan benar kondisi pasien sesungguhnya. Misalnya diperoleh nilai akurasi 95%, artinya jika ada pasien melakukan uji diagnosa maka hasil diagnosa tersebut 95% akurat atau tepat, baik pasien mempunyai kanker serviks dengan stadium 1,2,3 dan 4 atau tidak mempunyai kanker serviks (normal). Rumus akurasi adalah sebagai berikut :

$$Akurasi = \frac{jumlah\ diagnosis\ benar\ (cocok)}{jumlah\ diagnosis\ seluruhnya} \times 100\% \quad (2.59)$$



Your complimentary
use period has ended.
Thank you for using
PDF Complete.

Click Here to upgrade to
Unlimited Pages and Expanded Features

- Anami, B.S., Vishwanath C.B. 2009. Texture Based Identification and Classification of Bulk Sugary Food Object. *ICGST*. Vol. 9.
- Anita Desiani & Muhammad Arhami. (2006). *Konsep Kecerdasan Buatan*. Yogyakarta: Penerbit ANDI.
- Arif Munandar. (2014). *Metode Fuzzy-Two Step Filter untuk Mereduksi Impulse Noise pada Citra RGB (Red Green Blue) dan Aplikasinya pada Citra Fotografi*. Yogyakarta: Universitas Negeri Yogyakarta.
- Biran, A., & Breiner, M. (1999). *MATLAB 5 for engineers*, Addison: Wesley
- Boden, M. (2001). *A Guide To Recurrent Neural Network And Backpropagation*. Swedia:Halmstad University.
- Crawley, M. J. (2014). *Statistics: An Introducion Using R*. West Sussex: Jhon Wiley & Sons Ltd.
- Deri Edianto. (2006). Kanker Serviks. *Onkologi Ginekologi* Edisi 1. Jakarta: Yayasan Bina Pustaka Sarwono Prawirohardjo. Hlm: 442.
- Devi Nutriyastari . (2014). *Aplikasi Model Recurrent Neural Network dan Model Recurrent Neuro Fuzzy untuk Klasifikasi Nodul Kanker Paru dari Citra Foto Paru*. Yogyakarta: Universitas Negeri Yogyakarta.
- Dobrescu, R., Dobrescu, M., Mocanu, S., & Popescu, D. (2010). *Medical Image Classification for skin cancer diagnosis based on combined texture and fractal analysis*. Romania: Splaiul Independentei.
- Fauset, L. (1994). *Fundamental Of Neural Network*. Upper Saddle River: Prentice Hall Inc.
- Gadkari, D. (2000). *Image Quality Analysis Using GLCM*. Orlando: University of Central Florida.
- Gonzalez, R.C. & Woods, R.E. (2002). *Digital Image Processing 2nd Edition*. Upper Saddle River: Prentice Hall.

75

- Sankaranarayanan, R. & Ramani S.W. (2003). *A Practical Manual on Visual Screening for Cervical Neoplasia*. WHO: IARC.
- Sharma, M. & Mukherjee, S. (2014). Fuzzy C-Means, ANFIS, and Genetic Algorithm for Segmenting Astrocytoma-A type of Brain Tumor. *IAES International Journal of Artificial Intelligence*, Vol. 3, Hlm 16-23.
- Siang. (2005). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta: Penerbit Andi.
- Simundic, A.M. (2009). *Measures of Diagnostic Accuracy : Basic Definitions*. Zagreb : Sestre Milosrdnice University.
- Soh, L. & Tsatsoulis, C. (1999). Texture Analysis of SAR Sea Ice Imagery Using Gray Level Co-occurrence Matrices. *IEEE Transaction on Geoscience and Remote Sensing*, Vol 37, No2, Hlm. 780-795.
- Sri Kusumadewi & Sri Hartati. (2010). *Neuro-Fuzzy Integrasi Sistem Fuzzy dan Jaringan Syaraf*. Yogyakarta: Graha Ilmu.
- Sri Kusumadewi. (2004). *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK*. Yogyakarta: Graha Ilmu.
- Tim Kanker Serviks. 2010. *Panduan Lengkap Menghadapi Bahaya Kanker Serviks*. www.kanker-serviks.net/artikel.
- www.cancerhelps.com/kanker-serviks.htm diakses pada tanggal 1 Februari 2015.
- Yeffry Handoko, Jhon Adler, & Gugun Gunawan. (2013). *Aplikasi Filter Finite Impulse Response (FIR) untuk Menghilangkan Noise pada Suara Manusia Menggunakan Graphical User Interface (GUI) Pemrograman MATLAB*. Bandung : Unikom.
- Yushaila Nursajida. (2013). *Klasifikasi Stadium Kanker Serviks Menggunakan Model Fuzzy*. Yogyakarta : Universitas Negeri Yogyakarta.

77

- Haralick, R.M., Shanmugam, K. & Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transaction on System, Man and Cybernetics*, Vol.3, Hlm 610-621.
- Haykin, S. (1999). *Neural Network: A Comprehensive Foundation*. New York: Prentice Hall.
- I Dewa Gede Sukardja. (2000). *Onkologi Klinik* Edisi 2. Surabaya: Airlangga University Press.
- Kaastra, I., Boyd, M. (1996). *Designing a Neural Network for Forecasting Financial and Economic Time Series*. Amsterdam: Elsevier.
- Kanchanaraksa, S. (2008). *Evaluation of Diagnostic and Screening Tests: Validity and Reliability*. Washington : Jhon Hopkins University.
- Lin Chao, C et al, (2008). *Artificial Neural Network Prediction of Clozapine Response with Combine Pharmacogenetic and Clinical Data*.
- Lin, C.-T. & Lee, G. (1996). *Neuro Fuzzy System*. New Jersey: Prentice-Hall.
- McAndrew, A. 2004. *An Introduction to Digital Image Processing with MATLAB*. School of Computer Science and Mathematics : Victoria University Technology.
- McLeod, A.I. (1991). *Parsimony, Model Adequacy and Periodic Correlation in Time Series Forecasting*. Londen : The University of Western Ontario.
- Mohanaiah, P., Sathyanarana, P. & Guru, K.L. (2013). Image Texture Feature Extraction Using GLCM Approach. *International Journal of Scientific and Research Publication*, Vol. 3, Hlm 1-5.
- Muhammad Farid Aziz., Andrijono. & Abdul Bari Saifuddin. (2006). *Onkologi Ginekologi* Edisi 1. Jakarta: Yayasan Bina Pustaka Sarwono Prawirohardjo.
- Rama Diananda. (2007). *Mengenal Seluk-Beluk Kanker*. Yogyakarta: Ar-Ruzz Media.
- Reni Setyaningrum. (2014). *Klasifikasi Stadium Kanker Kolorektal Menggunakan Model Neural Network*. Yogyakarta : Universitas Negeri Yogyakarta.
- Salini, R., Bristow, R.E. (2011). *Jhons Hopkins Medicine. Panduan untuk Penderita Kanker Ovarium*. Jakarta: PT Indeks.
- Sankaranarayanan, R. & John W.S. (2003). *Colposcopy and Treatment of Cervical Intraepithelial Neoplasia*. WHO: IARC.

76